

2 Отложенные вычисления и ленивые списки в Лиспе. Нисходящий разбор по заданной LL(1)-грамматике

Введение. Нисходящий синтаксический анализ осуществляется автоматом с магазинной памятью (МП-автоматом). В начале работы МП-автомата в стек помещается стартовый нетерминал. Далее автомат недетерминировано заменяет нетерминалы в стеке на правые части соответствующих продукций, а терминалы снимает с вершины стека вместе с чтением их из входной строки. Строка допускается, если по окончании чтения строки стек пуст. Недетерминизм на практике реализуется сложно, поэтому там, где это возможно, грамматики приводятся к специальной форме, позволяющей от него избавиться. Одна из таких форм — LL(1). Первая «L» означает чтение входной строки слева направо, вторая «L» — получение левого порождения, а «1» — использование на каждом шаге предпросмотра одного символа для принятия решения о действиях автомата.

Грамматика G называется LL(1)-грамматикой, если для любых двух различных продукций $A \rightarrow \alpha \mid \beta$ выполняются следующие условия:

1. $\text{first } \alpha \cap \text{first } \beta = \emptyset$.
2. Если β порождает пустую строку, то $\text{first } \alpha \cap \text{follow } A = \emptyset$. Аналогично, если α порождает пустую строку, то $\text{first } \beta \cap \text{follow } A = \emptyset$.

$\text{first } \alpha$ — это множество, содержащее все терминалы, с которых начинаются строки, выводимые из α и ε , если α порождает пустую строку.

$\text{follow } A$ — множество терминалов, которые могут располагаться непосредственно справа от A в некоторой сентенциальной форме.

Задание. Написать программу, эмулирующую поведение детерминированного МП-автомата, составленного по заданной LL(1)-грамматике. Проверку принадлежности строки языку выполнить в виде поиска допускающего состояния в истории вычислений автомата. Историю вычислений реализовать в виде отложенного списка, то есть не допускается одновременное хранение в памяти всех промежуточных состояний автомата.

Символы, не встречающиеся в левых частях грамматик, и считать терминалами.

1. $\text{expr} \rightarrow \text{term expr2}$
 $\text{expr2} \rightarrow + \text{term expr2} \mid \varepsilon$
 $\text{term} \rightarrow \text{fact term2}$
 $\text{term2} \rightarrow * \text{fact term2} \mid \varepsilon$
 $\text{fact} \rightarrow (\text{expr}) \mid a$

2. $stmt \rightarrow if (expr) stmt \text{ else } stmt \mid while (expr) stmt \mid \{stmtlist\}$
 $stmtlist \rightarrow stmt \text{ stmtlist} \mid \varepsilon$
3. $stmt \rightarrow if \text{ expr then } stmt \text{ else } stmt$
 $expr \rightarrow term \text{ expr2}$
 $expr2 \rightarrow or \text{ term expr2} \mid and \text{ term expr2} \mid \varepsilon$
 $term \rightarrow a$
4. $proc \rightarrow head \text{ proclist body}$
 $head \rightarrow procedure \text{ name (params)}$
 $proclist \rightarrow proc \text{ proclist2}$
 $proclist2 \rightarrow ; \text{ proc proclist2} \mid \varepsilon$
 $params \rightarrow param \text{ params2}$
 $params2 \rightarrow , \text{ param params2} \mid \varepsilon$